

AD-A035 363

TEXAS UNIV AT AUSTIN DEPT OF ELECTRICAL ENGINEERING
FOURIER SERIES REPRESENTATION FOR POLYNOMIALS WITH APPLICATION --ETC(U)
1976 N C GALLAGHER, J ALLEN, G L WISE

AF-AFOSR-3062-76

AFOSR-TR-77-0023

F/G 12/1

NL

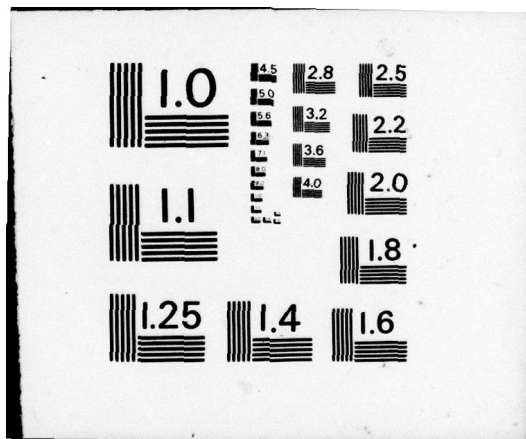
UNCLASSIFIED

1 OF 1
ADA035363



END

DATE
FILMED
3-77



ADA035363

FOURIER SERIES REPRESENTATION FOR POLYNOMIALS WITH APPLICATION TO NONLINEAR DIGITAL FILTERING

NEAL C. GALLAGHER, JR.
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

JOHN W. ALLEN
Department of Electrical Engineering & Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106

GARY L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

Handwritten signature and circled number 4.

DDC
RECEIVED
FEB 9 1977
D

ABSTRACT

A change-of-basis transformation between the trigonometric Fourier Series and the Legendre polynomial series is derived. This transformation leads to two immediate results: one is a fast algorithm for computing the Legendre polynomial coefficients for a function; the second is a numerically stable computation algorithm for evaluating $j_n(z)$, the n^{th} order spherical Bessel function. In addition, these results lead to a simple recursive digital network to be used in the implementation of adaptive nonlinear functions.

I. INTRODUCTION

Two common ways of representing functions have been polynomial expansions and trigonometric expansions. In much of electrical engineering the trigonometric expansion has useful interpretations and has dominated over the generalized Fourier series expansions in applications. However, many functions are readily expressed in terms of polynomials. For example, in a recent paper, we have presented an interesting representation for bandlimited functions using the Legendre polynomial expansion [1]. In this paper we derive a simple linear transformation which maps the polynomial representation into a trigonometric representation. Also, we derive the inverse transformation which maps a trigonometric expansion to a polynomial expansion.

The inverse transformation has enabled us to develop a fast algorithm for the computation of the Legendre polynomial coefficients for any $L_2[-\pi, \pi]$ function. The algorithm utilizes the Fast Fourier Transform (FFT) to compute the Fourier series coefficients and then multiplies the vector of coefficients by a linear matrix transformation to compute the vector of polynomial coefficients. This approach can offer a considerable saving in computation time over the standard integral formula for computing these polynomial coefficients.

Section II contains the derivation for the elements of the transformation matrix and its inverse. In section III we discuss how these results lead to the design of a simple recursive digital filter that may be employed to implement almost any zero memory non-linear operation (ZNL). In section IV we discuss several computational considerations, and in section V we present an example.

II. DEVELOPMENT

Assume $H(x)$ is a polynomial defined on $[-\pi, \pi]$. By writing $H(x)$ in Legendre polynomials, we can use results developed in an earlier paper to easily write the Fourier series expansion for $H(x)$ that converges in $[-\pi, \pi]$.

Presented at the Fourteenth Annual Allerton Conference on Circuit and System Theory, 1976; to be published in the Proceedings of the Conference.

Approved for public release;
distribution unlimited.

Assume that

$$H(x) = \sum_{n=0}^N a_n P_n\left(\frac{x}{\pi}\right), \quad -\pi \leq x \leq \pi \quad (1)$$

where $P_n(\cdot)$ is the n th Legendre polynomial. To determine the Fourier series representation for $H(x)$, it is sufficient to know the Fourier series expansion for $P_n\left(\frac{x}{\pi}\right)$. From [1] we know that

$$P_n\left(\frac{x}{\pi}\right) = \sum_{m=-\infty}^{\infty} [(i)^n j_n(m\pi)] e^{-imx}. \quad (2)$$

By Eqs. (1) and (2), we can write

$$H(x) = \sum_{n=-\infty}^{\infty} h_n e^{-inx}, \quad (3a)$$

where

$$h_n = \sum_{m=0}^N a_m (i)^m j_m(n\pi). \quad (3b)$$

We note that the sequence $\{h_n\}$ can be treated as a matrix multiplied by a vector (in an infinite dimensional vector space). That is,

$$h = a B,$$

where a is a row vector of polynomial coefficients

$$a = [a_0, a_1, \dots, a_N]$$

and B is the $(N+1) \times \infty$ transformation matrix whose (m,n) th element is given by

$$b_{mn} = (i)^m j_m(n\pi).$$

Observe that the transformation matrix B is independent of the function $H(x)$, and therefore a different transformation matrix need not be computed for each different $H(x)$. A procedure for the computation of the terms $j_m(n\pi)$ is contained in section IV.

Now we consider the case where $H(x)$ is not a polynomial. Assume that $H(x)$ belongs to $L_2[-\pi, \pi]$, and therefore possesses a Legendre polynomial series expansion convergent in $L_2[-\pi, \pi]$. The coefficient vector becomes infinite dimensional, and the transformation matrix B is doubly infinite. As a result, Eq. (3b) is written as

$$h_n = \sum_{m=0}^{\infty} a_m (i)^m j_m(n\pi). \quad (4)$$

It can be shown [1] that the sum in Eq. (4) converges uniformly in n . That is, for any $\epsilon > 0$, there exists a K such that for all $M \geq K$

$$|h_n - h_n(M)| < \epsilon$$

for all n , where

$$h_n(M) = \sum_{m=0}^M a_m (i)^m j_m(n\pi).$$

Thus the sequence $\{h_n(M)\}$ is a uniform approximation to the sequence of Fourier series coefficients for the function $H(x)$.

Now, we consider the inverse transformation denoted by B^{-1} . The computation of B^{-1} directly from the matrix B can be performed; however, it is

best to compute the elements of B^{-1} in a fashion similar to the computation of the elements of B . First, we note that the coefficients $\{a_n\}$ in Eq. (1) are given by the relation [1]

$$a_n = \frac{2n+1}{2\pi} \int_{-\pi}^{\pi} H(x) P_n\left(\frac{x}{\pi}\right) dx \quad (6)$$

Using the Fourier series expansion for $H(x)$ found in Eq. (3a), we find

$$a_n = (2n+1) \sum_{m=-\infty}^{\infty} h_m \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} P_n\left(\frac{x}{\pi}\right) e^{-imx} dx \right] \quad (7)$$

Arguing as in the derivation of Eq. (2), we have

$$a_n = (2n+1) \sum_{m=-\infty}^{\infty} h_m [(-i)^n j_n(m\pi)] \quad (8)$$

If we define

$$\hat{b}_{mn} = (2n+1) (-i)^n j_n(m\pi), \quad (9a)$$

then

$$[\hat{b}_{mn}] = B^{-1}; \quad (9b)$$

that is, the matrix elements of B^{-1} are given by \hat{b}_{mn} . Consequently, Eq. (8) may be rewritten as

$$a = h B^{-1}, \quad (10)$$

where h is the row vector of Fourier series coefficients and a is the row vector of polynomial coefficients. In practice, a finite number of elements of h are computed by use of the FFT algorithm; we then perform the vector multiplication indicated by Eq. (10) to compute a finite number of elements of the vector a . This procedure for computing the Legendre polynomial coefficients for a function can provide a great reduction in computation time as compared to the direct evaluation of the integral in Eq. (6).

In the next section, we discuss a technique for the implementation of nonlinear operations by use of a simple recursive digital structure.

III. DIGITAL ZNL IMPLEMENTATION

We begin with the following recursion relation for Legendre Polynomials

$$(n+1) P_{n+1}(x) = (2n+1) x P_n(x) - n P_{n-1}(x), \quad (11)$$

where

$$P_0(x) = 1$$

$$P_1(x) = x, \quad -1 \leq x \leq 1.$$

It is straight forward to implement this relation with a recursive digital network. Then, given this method by which to generate $\{P_n(x)\}$ for a given x , and given the coefficients $\{a_n\}$, the nonlinear function $H(x)$ is easily constructed as

$$H(\pi x) \approx \sum_{n=0}^N a_n P_n(x), \quad -1 \leq x \leq 1. \quad (12)$$

Figure 1 contains one possible implementation of Eqs. (11) and (12). In Fig. 1 a recursive network is used to generate $P_n(x)$ for successive values

of n . With the coefficients $\{a_n\}$ stored in memory, the products and sum indicated by Eq. (12) are then performed. We note that the function $H(x)$ implemented by this network may be easily changed simply by changing the coefficients $\{a_n\}$; consequently, except for a memory unit, the network remains unchanged regardless of the function $H(x)$.

Suppose that in order to obtain acceptable accuracy at the system output it is necessary, to make $N = 50$ in Eq. (12). Thus, given a value x we must wait for 50 terms to be accumulated before we have the output $H(x)$. Therefore this digital ZNL could reduce the system data rate by a factor of 50; however, for a price, it is possible to eliminate this rate reduction. One approach would be to simply use a faster clock rate with the ZNL structure. Another approach is to construct $N = 50$ such ZNL structures; the system's first input sample, call it x_1 , goes into the first ZNL; the second sample, x_2 , goes into the second ZNL and so on. Now, just as x_{50} is going into the 50th ZNL, we are getting the output $H(x_1)$ from the first ZNL. Consequently, after an initial delay of 50 samples, 50 separate parallel networks can operate at a rate 50 times greater than a single network. Clearly, there are compromises that can be made between hardware cost and data rate.

Potential applications of this ZNL structure include adaptive nonlinear systems. In such an application the coefficients $\{a_n\}$ can be continually updated by use of the fast algorithms previously discussed.

IV. COMPUTATIONAL CONSIDERATIONS

In this section we discuss two different methods for the computation of $j_n(m\pi)$. We begin by noting the following relation for generating values of $j_n(z)$ [2]

$$j_n(z) = f_n(z) \sin z + (-1)^{n+1} f_{-n-1}(z) \cos z, \quad (13)$$

where

$$f_0(z) = \frac{1}{z}, \quad f_1(z) = z^{-2}$$

and

$$f_{n-1}(z) + f_{n+1}(z) = (2n+1) z^{-1} f_n(z).$$

Consequently,

$$j_n(m\pi) = (-1)^{n+1+m} f_{-n-1}(m\pi), \quad n = 1, 2, \dots$$

where

$$j_0(m\pi) = \begin{cases} 1, & m = 0 \\ 0, & \text{otherwise} \end{cases}$$

The recursion in Eq. (13) appears to provide a relatively simple method for computing $j_n(m\pi)$; unfortunately, this relation is numerically unstable. We have found that even with double precision arithmetic error accumulates rapidly. When using this recursion relation, we have produced overflow conditions for $n = 20$ and $z = \pi, 2\pi$ on a UNIVAC 1108 digital computer; this is, of course, unacceptable.

Fortunately, we have developed a numerical technique that appears to be very stable. First, we note that the recursion relation for generating $P_n(x)$ in Eq. (11) is very tolerant of roundoff accumulation. Then, we observe that Eq. (2) implies the following approximate relationship.

DESIGN for	
WTS	White Section <input checked="" type="checkbox"/>
ESG	Defi Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
DISTRIBUTION/AVAILABILITY CODES	
SPECIAL and/or SPECIAL	
A	

$$P_n\left(\frac{2q}{N}\right) = \sum_{m=-\frac{N}{2}+1}^{N/2} \left[(i)^n j_n(m\pi) \right] e^{-i \frac{2\pi m q}{N}} \quad q = -\frac{N}{2} + 1, \dots, \frac{N}{2} \quad (14)$$

which is the discrete Fourier transform relation. Consequently, we have the inverse relationship

$$(i)^n j_n(m\pi) = \frac{1}{N} \sum_{q=-\frac{N}{2}+1}^{N/2} P_n\left(\frac{2q}{N}\right) e^{i \frac{2\pi m q}{N}}, \quad m = -\frac{N}{2} + 1, \dots, \frac{N}{2} \quad (15)$$

Except for aliasing error, Eq. (15) is accurate. In order to reduce aliasing error it is necessary to increase N ; generally speaking, N must be chosen separately for each case, depending on the degree of the polynomial $P_n(\cdot)$. Our procedure for computing $j_n(m\pi)$ is as follows: using Eq. (11), values of $P_n\left(\frac{2q}{N}\right)$ are generated; then, using Eq. (15), values of $j_n(m\pi)$ are computed. The values of $j_n(m\pi)$ used in the following section's example have been computed by this method. We now discuss factors considered in the computation of the coefficients $\{a_n\}$.

If we assume that $H(x)$ is real-valued, then Eq. (8), used to evaluate a_n , can be simplified by noting that the Fourier coefficients $\{h_m\}$ satisfy the relation $h_{-m} = h_m^*$. Also, $j_n(m\pi)$ satisfies the relation [2]

$$j_n(-m\pi) = (-1)^n j_n(m\pi). \quad (16)$$

Consequently, Eq. (8) may be rewritten as

$$\begin{aligned} a_0 &= h_0 \\ a_n &= 2(2n+1)(-1)^{\frac{n}{2}} \sum_{m=1}^{\frac{\hat{N}}{2}} j_n(m\pi) \operatorname{Re} \{h_m\}, \quad n\text{-even and } \neq 0 \\ a_n &= 2(2n+1)(-1)^{\frac{n-1}{2}} \sum_{m=1}^{\frac{\hat{N}}{2}} j_n(m\pi) \operatorname{Im} \{h_m\}, \quad n\text{-odd,} \end{aligned} \quad (17)$$

where $\operatorname{Re} \{h_m\}$ and $\operatorname{Im} \{h_m\}$ denote the real and imaginary parts of h_m , respectively; and the maximum value for \hat{N} is N , which is the number of coefficients $\{h_m\}$ that are computed. By use of Eq. (17), the polynomial coefficients $\{a_n\}$ are computed from the Fourier coefficients $\{h_n\}$.

The following section contains a numerical example computed with the aid of the results summarized in Eqs. (15) and (17).

V. AN EXAMPLE

In this section we will demonstrate the computation of the polynomial coefficients $\{a_n\}$ for

$$\begin{aligned} H(\pi x) &= P_3(x) + P_2(x) \\ &= \frac{1}{2} (5x^3 + 3x^2 - 3x - 1), \quad -1 \leq x \leq 1. \end{aligned}$$

By comparison with Eq. (1), we see that all $a_n = 0$ with the exception of $a_2 = a_3 = 1$. For this example, all computations are single precision arithmetic. The Fourier coefficients $\{h_m\}$ are computed by use of the DFT

relation:

$$h_m = \frac{1}{N} \sum_{p=-\frac{N}{2}+1}^{\frac{N}{2}} H\left(\frac{2\pi p}{N}\right) e^{i \frac{2\pi m p}{N}}, \quad m = -\frac{N}{2} + 1, \dots, \frac{N}{2},$$

and the terms $\{j_n(m\pi)\}$ are computed by use of Eq. (15). Finally, the coefficients $\{a_n\}$ are computed by application of Eq. (17). These computations are performed for three different values of $N = \hat{N}$: $N = \hat{N} = 8192$, 4096, and 2048. Table 1 contains typical values of $\{a_n\}$.

N	2048	4096	8192	True Value
a_0	-4.880×10^{-4}	-2.441×10^{-4}	-1.220×10^{-4}	0
a_1	-1.463×10^{-3}	-7.320×10^{-4}	-3.661×10^{-4}	0
a_2	0.9976	0.9988	0.9994	1
a_3	0.9966	0.9983	0.9991	1
a_4	-4.385×10^{-3}	-2.195×10^{-3}	-1.098×10^{-3}	0
a_5	-5.353×10^{-3}	-2.681×10^{-3}	-1.342×10^{-3}	0

Table 1. Selected Values of $\{a_n\}$ Computer for $N = \hat{N}$.

As the value of N is increased the effect of aliasing is reduced in our computation of $\{j_n(m\pi)\}$ and $\{h_m\}$. Also, as the value of \hat{N} is increased, the summation terms of Eq. (17) become more accurate. Inspection of Table 1 reveals that as the value of $N = \hat{N}$ is doubled, the error in computation is halved. We ask, to which parameter, N or \hat{N} , is the error most sensitive? Some insight may be obtained by examining the value of a_0 in Table 1; the value of a_0 is independent of the value of \hat{N} (see Eq. 17). Because the computation error for a_0 is halved as N doubles, the implication is that much of the error is due to aliasing and is dependent on N . This is illustrated in Table 2, where the parameter \hat{N} is held constant at $\hat{N} = 2048$.

N	2048	4096	8192	True Value
a_0	-4.880×10^{-4}	-2.441×10^{-4}	-1.220×10^{-4}	0
a_1	-1.463×10^{-3}	-8.322×10^{-4}	-6.545×10^{-4}	0
a_2	0.9976	0.9988	0.9994	1
a_3	0.9966	0.9981	0.9985	1
a_4	-4.385×10^{-3}	-2.192×10^{-3}	-1.096×10^{-3}	0
a_5	-5.353×10^{-3}	-3.049×10^{-3}	-2.399×10^{-3}	0

Table 2. Selected Values of $\{a_n\}$ for $\hat{N} = 2048$.

First note that in Tables 1 and 2, the values of $\{a_n\}$ for $N = 2048$ are identical because the same value of $\hat{N} = 2048$ is used in both cases. Upon comparing the two tables, we note that the value of \hat{N} does affect the computation error; for some value $\{a_n\}$ the effect is negligible while with others a factor of 2 difference is noted.

The results of this section indicate the importance of computing the elements of the B^{-1} matrix very precisely, specifically $\{j_n(m\pi)\}$. If this computation is accomplished with the DFT procedure outlined previously, then a large value of N should be chosen for the DFT; we emphasize that this

computation need be made only once, and the results stored in memory.

V. DISCUSSION AND SUMMARY

In this paper, we have developed a fast algorithm for the computation of the Legendre polynomial coefficients of a function. Combining this fast algorithm with a digital implementation of the Legendre polynomial recursion relation, it is possible to implement an adaptive nonlinearity. Generally, the more complicated the non-linearity to be implemented, the higher will be the degree of the polynomial approximation required; this in turn, reduces the throughput rate of the non-linear device unless we are willing to pay a greater price in hardware. Nevertheless, by using this approach one acquires a large degree of flexibility in specifying the characteristics of the nonlinearity.

In the past, most non-linear devices used in practice have had very simple characteristics; in part, this has been due to the relative difficulty in implementing more complicated devices. However, as signal processing schemes become increasingly sophisticated, the need for complex or adaptive non-linear devices is growing. One very interesting application is in robust or distribution free signal detection schemes [3] where an adaptive, easily implementable, non-linearity can be of great benefit.

In addition to its application to non-linear systems, we should not overlook the advantage of having a fast algorithm for the computation of the polynomial expansion for almost any function.

It should also be noted that there are many special function pairs with similar Fourier transform relationships that may be exploited in a fashion similar to that presented in this paper [4]. The fact that there may not be a simple expression for computing B^{-1} should not be a deterrent from performing the computations because B^{-1} need only be computed once and then stored in memory for all future requirements.

ACKNOWLEDGEMENT

The support of the National Science Foundation under Grant ENG 76-09443, and of the Air Force Office of Scientific Research under Grant AFOSR-76-3062 is gratefully acknowledged.

REFERENCES

1. N. C. Gallagher and G. L. Wise, "A Representation for Band-Limited Functions," Proc. of the IEEE, Vol. 63, pp. 1624-1625, Nov. 1975.
2. M. Abramowitz and I. A. Segun, Handbook of Mathematical Functions, Dover Publications, New York, N.Y., p. 782.
3. S. A. Kassam and S. B. Thomas, "Generalizations of the Sign Detector Based on Conditional Tests," IEEE Trans. on Communications, Vol. COM-24, pp. 481-487, May 1976.
4. K. Yao and J. B. Thomas, "On Band-Limited Properties of Fourier Transform Pairs of Some Special Functions," Proc. 3rd Annu. Allerton Conf. on Circuit and System Theory, pp. 299-309, Oct. 1965.

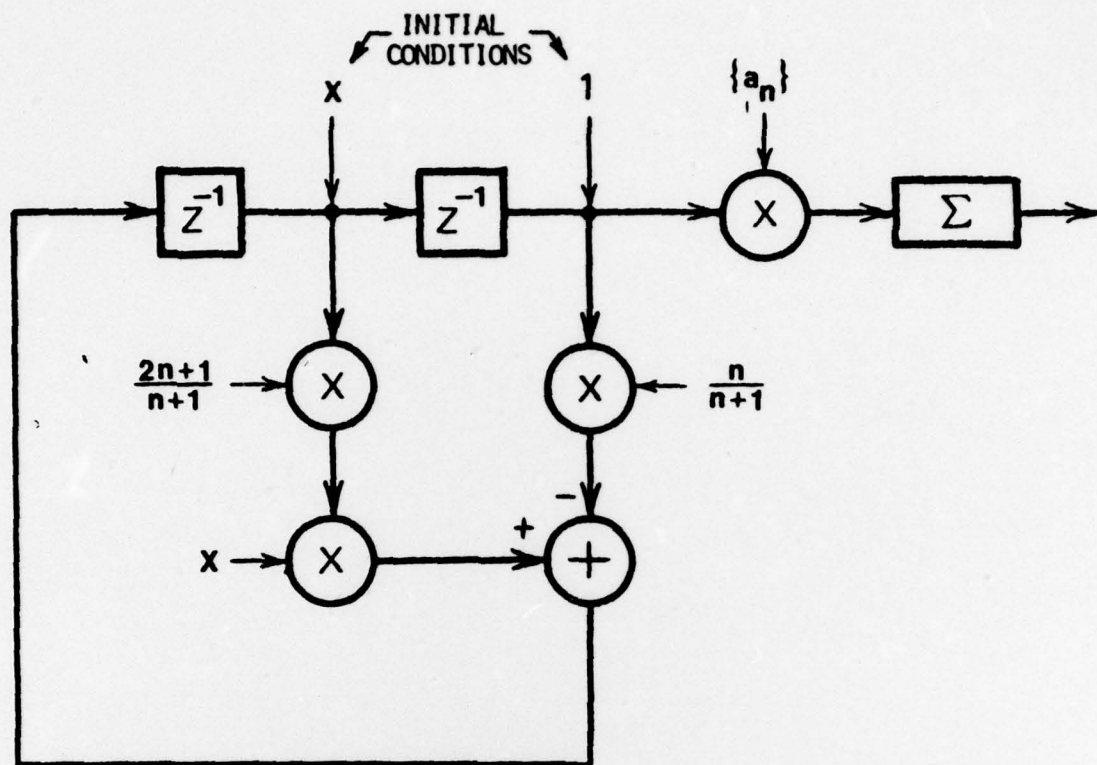


Fig. 1. Digital Network for the Computation of $\sum a_n P_n(x)$.

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER AFOSR - TR-77-0023	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) FOURIER SERIES REPRESENTATION FOR POLYNOMIALS WITH APPLICATION TO NONLINEAR DIGITAL FILTERING		5. TYPE OF REPORT & PERIOD COVERED Interim	
7. AUTHOR(s) Neal C. Gallagher, John Allen Gary L. Wise		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS The University of Texas at Austin Department of Electrical Engineering Austin, Texas 78712		8. CONTRACT OR GRANT NUMBER(s) 15 VAF-AFOSR-3062-76	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, Washington, DC 20332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 16 2314 17 A5	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 129 P.		12. REPORT DATE 11/1976	
		13. NUMBER OF PAGES 8	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES Presented at the Fourteenth Annual Allerton Conference on Circuit and System Theory, 1976; to be published in the Proceedings of the Conference.			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) polynomial expansions Legendre polynomials			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A change-of-basis transformation between the trigonometric Fourier Series and the Legendre polynomial series is derived. This transformation leads to two immediate results: one is a fast algorithm for computing the Legendre polynomial coefficients for a function; the second is a numerically stable computation algorithm for evaluating $j_n(z)$, and n th order spherical Bessel function. In addition, these results lead to a simple recursive digital network to be used in the implementation of adaptive nonlinear functions. j sub n			